

1. В базе данных Redis подберите коллекцию для подсчета посещений с определенных IP-адресов.

Вариант 1 (нас интересуют уникальные значения ip)

Т.к. нас интересуют уникальные значения ip, то для хранения используем множество

```
flash@flash-VirtualBox:~$ redis-cli
127.0.0.1:6379> SADD ip '127.0.0.2' '127.0.0.3' '127.0.0.4'
(integer) 3
127.0.0.1:6379> SADD ip '127.0.0.2'
(integer) 0
127.0.0.1:6379> SADD ip '127.0.0.1'
(integer) 1
127.0.0.1:6379> SMEMBERS ip
1) "127.0.0.3"
2) "127.0.0.1"
3) "127.0.0.4"
4) "127.0.0.2"
127.0.0.1:6379> SCARD ip
(integer) 4
```

Вариант 2 (подсчитываем количество посещений с определенного ip)

В этом случае каждый ip будет являться ключом, значение увеличивается при каждом посещении

```
flash@flash-VirtualBox:~$ redis-cli
127.0.0.1:6379> set 127.0.0.1 0
OK
127.0.0.1:6379> set 127.0.0.2 0
OK
127.0.0.1:6379> set 127.0.0.3 0
OK
127.0.0.1:6379> set 127.0.0.4 0
OK
127.0.0.1:6379> incr 127.0.0.1
(integer) 1
127.0.0.1:6379> incr 127.0.0.1
(integer) 2
127.0.0.1:6379> incr 127.0.0.2
(integer) 1
127.0.0.1:6379> incr 127.0.0.3
(integer) 1
127.0.0.1:6379> get 127.0.0.1
"2"
```

2. При помощи базы данных Redis решите задачу поиска имени пользователя по электронному адресу и наоборот, поиск электронного адреса пользователя по его имени.

Если правильно понял задание, то создаем пары значений имя-емэйл и наоборот емэйл-имя, т.к. поиск может быть только по ключу

```
127.0.0.1:6379> set sveta@mail.ru sveta
OK
127.0.0.1:6379> set sveta sveta@mail.ru
OK
127.0.0.1:6379> set roma romash123@mail.ru
OK
127.0.0.1:6379> set romash123@mail.ru roma
OK
127.0.0.1:6379> get roma
"romash123@mail.ru"
127.0.0.1:6379> get sveta
"sveta@mail.ru"
127.0.0.1:6379> get sveta@mail.ru
"sveta"
127.0.0.1:6379> get romash123@mail.ru
"roma"
```

3. Организуйте хранение категорий и товарных позиций учебной базы данных shop в СУБД MongoDB.

В ДБ products дублируем значения из shop.products, каждая позиция - объект

```
> use products
switched to db products
> db.products.insert({"name": "Intel Core i3-8100", "description": "Процессор Intel", "price": "7890.00", "catalog_id": "Процессоры", "created_at":
new Date(), "updated_at": new Date()})

WriteResult({ "nInserted" : 1 })
> db.products.insertMany([
... {"name": "AMD FX-8320", "description": "Процессор AMD", "price": "7120.00", "catalog_id": "Процессоры", "created_at": new Date(), "updated_at":
new Date()},
... {"name": "MSI B250M GAMING PRO", "description": "Мат. плата, B250, Socket 1151, DDR4, mATX", "price": "5060.00", "catalog_id": "Мат. платы", "cr
eated_at": new Date(), "updated_at": new Date()}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5f2ee575a5d48e026273129d"),
    ObjectId("5f2ee575a5d48e026273129e")
  ]
}
> db.products.find().pretty()
{
  "_id" : ObjectId("5f2ee463a5d48e026273129c"),
  "name" : "Intel Core i3-8100",
  "description" : "Процессор Intel",
  "price" : "7890.00",
  "catalog_id" : "Процессоры",
  "created_at" : ISODate("2020-08-08T17:44:03.313Z"),
  "updated_at" : ISODate("2020-08-08T17:44:03.313Z")
}
{
  "_id" : ObjectId("5f2ee575a5d48e026273129d"),
  "name" : "AMD FX-8320",
  "description" : "Процессор AMD",
  "price" : "7120.00",
  "catalog_id" : "Процессоры",
  "created_at" : ISODate("2020-08-08T17:48:37.381Z"),
  "updated_at" : ISODate("2020-08-08T17:48:37.381Z")
}
{
  "_id" : ObjectId("5f2ee575a5d48e026273129e"),
  "name" : "MSI B250M GAMING PRO",
  "description" : "Мат. плата, B250, Socket 1151, DDR4, mATX",
  "price" : "5060.00",
  "catalog_id" : "Мат. платы",
  "created_at" : ISODate("2020-08-08T17:48:37.381Z"),
  "updated_at" : ISODate("2020-08-08T17:48:37.381Z")
}
```

Поиск по названию

```
> db.products.find({name: "MSI B250M GAMING PRO"}).pretty()
{
  "_id" : ObjectId("5f2ee575a5d48e026273129e"),
  "name" : "MSI B250M GAMING PRO",
  "description" : "Мат. плата, B250, Socket 1151, DDR4, mATX",
  "price" : "5060.00",
  "catalog_id" : "Мат. платы",
  "created_at" : ISODate("2020-08-08T17:48:37.381Z"),
  "updated_at" : ISODate("2020-08-08T17:48:37.381Z")
}
```

В ДБ catalogs хранятся названия категорий

```
> use catalogs
switched to db catalogs
> db.catalogs.insertMany([{"name": "Процессоры"}, {"name": "Мат.платы"}, {"name": "Видеокарты"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5f2ee6aca5d48e026273129f"),
    ObjectId("5f2ee6aca5d48e02627312a0"),
    ObjectId("5f2ee6aca5d48e02627312a1")
  ]
}
```